

第5章 数组和广义表

DATA STRUCTURE

计算机科学学院 廖雪花

本章内容简介

数组和广义表

5.1 数组的定义

5.2 数组的顺序表示和实现

5.3 矩阵的压缩存储

5.4 广义表的定义

5.5 广义表的存储结构

5.5 广义表的存储结构

廖雪花 LiaoXuehua

广义表的存储结构

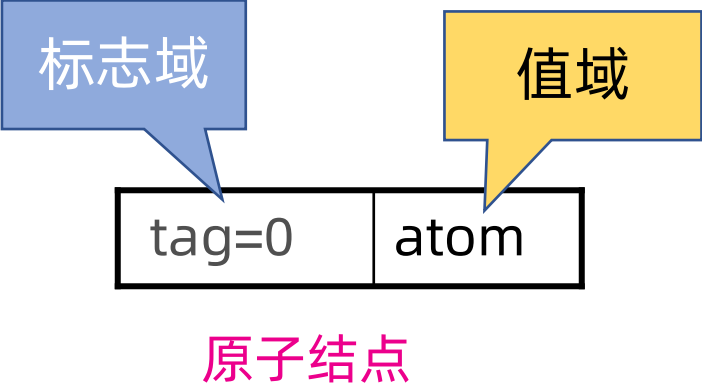
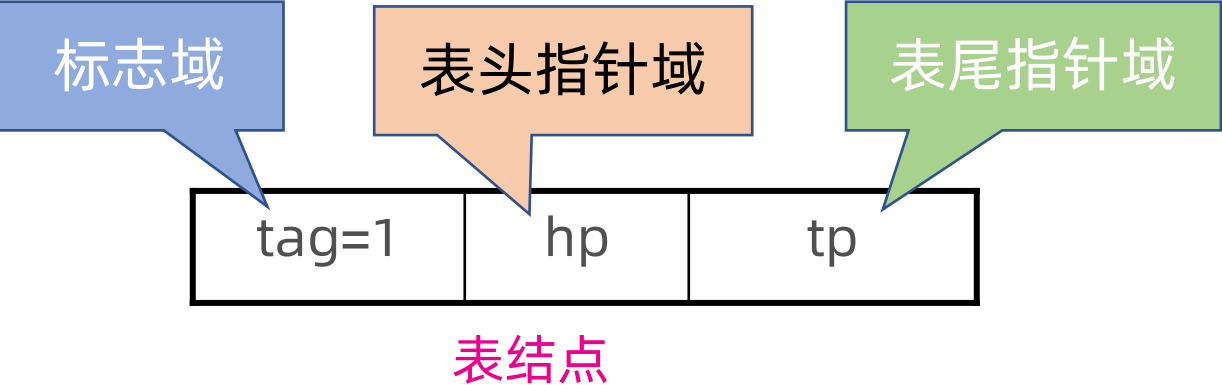
■ 存储结构的选择

由于广义表中的数据元素可以具有不同的结构（或是原子，或是列表），因此难以用顺序存储结构表示，通常采用链式存储结构，每个数据元素可用一个结点表示。



如何设定结点的结构呢？

广义表的存储结构



广义表的存储结构

■ 一、头尾链表存储结构

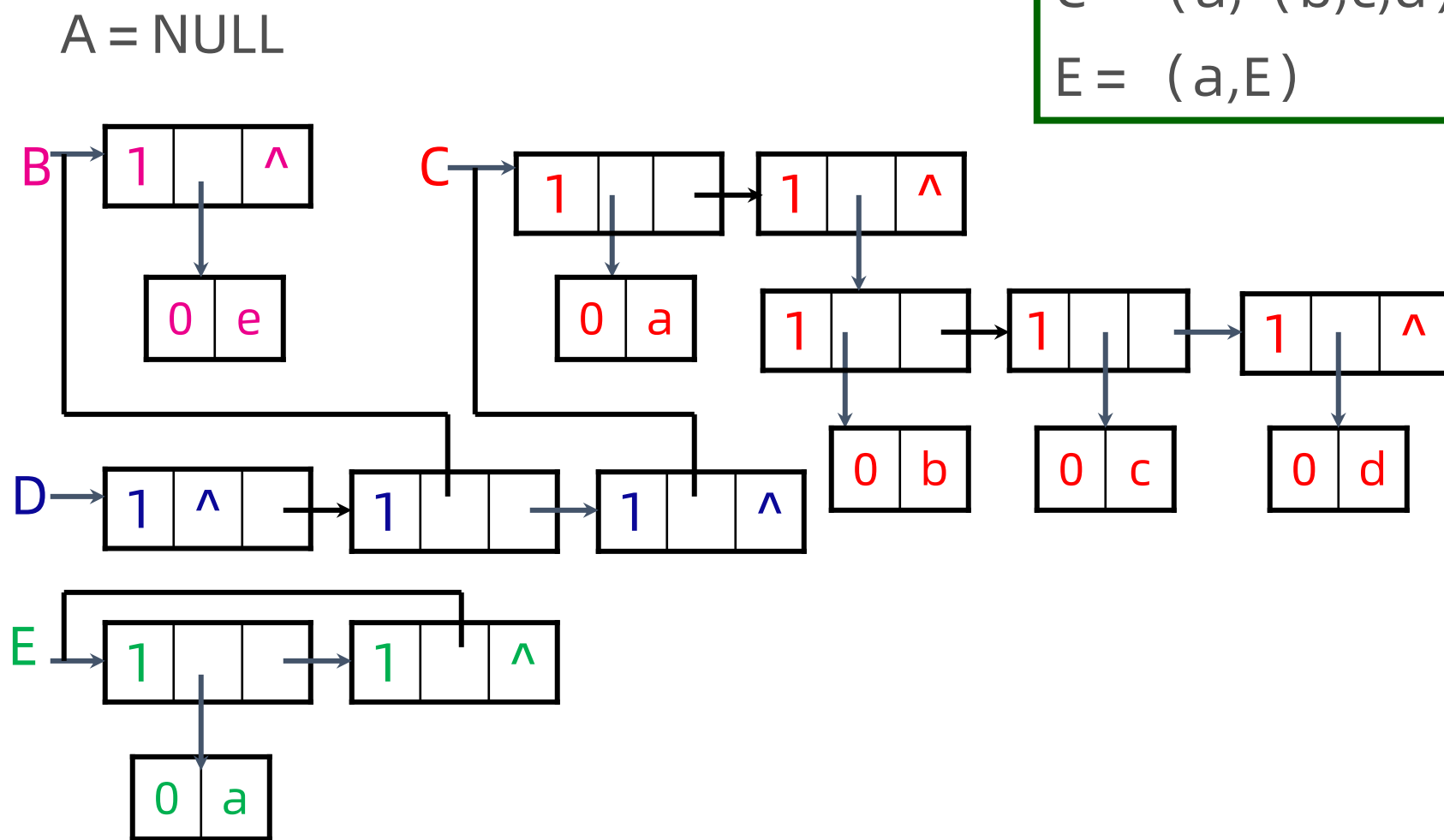
```
typedef enum{ATOM,LIST} ElemTag;  
typedef struct GLNode {  
    ElemTag tag;  
    union{  
        AtomType atom;  
        struct {struct GLNode *hp,*tp;} ptr;  
    };  
}*GList;
```

• 示例:

A = () B = (e)

C = (a, (b, c, d)) D = (A, B, C)

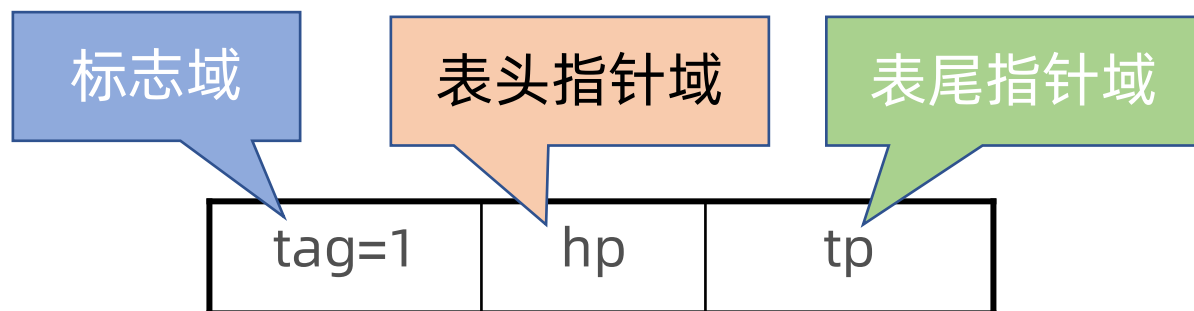
E = (a, E)



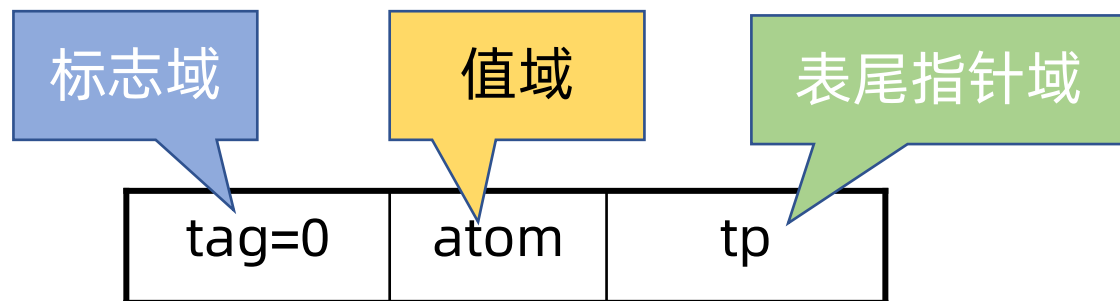
- 特点:

- (1) 若广义表为空，则表头指针为空；否则表头指针始终指向一个表结点。
- (2) 求表的长度、深度、取表头和表尾运算十分方便。
- (3) 表结点太多，造成浪费。

二、广义表的另一种链表存储结构



表结点



原子结点

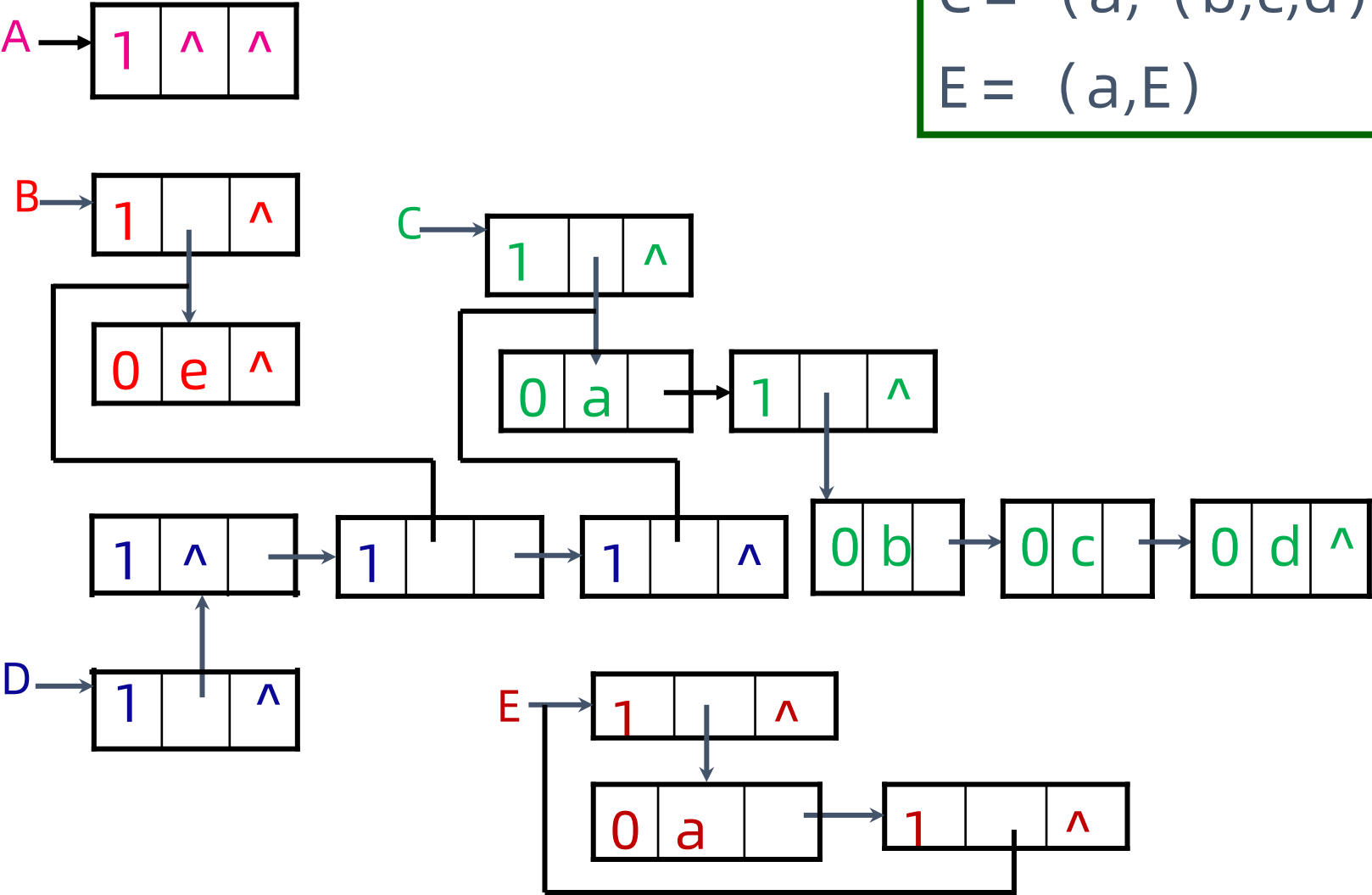
二、广义表的另一种链表存储结构

■ 扩展线性链表存储结构

```
typedef enum{ATOM,LIST} ElemTag;
typedef struct GLNode {
    ElemTag tag;
    union{
        AtomType atom;
        struct GLNode *hp;
    };
    struct GLNode *tp;//指向下一元素结点
}*GList;
```

举例：

A = () B = (e)
C = (a, (b,c,d)) D = (A,B,C)
E = (a,E)



广义表的应用

- m元多项式的表示

- 广义表的递归算法

本节要点

- 广义表的存储结构：
 - ✓ 头尾链表存储结构
 - ✓ 扩展线性链表存储结构

本章小结

- 数组的逻辑结构和顺序存储结构
- 矩阵的压缩存储
 - ◆ 特殊矩阵
 - ◆ 稀疏矩阵压缩存储方法及有关运算的实现
- 广义表
 - ◆ 是一种复杂的非线性结构，是线性表的推广
 - ◆ 本章简要介绍了它的定义、基本运算和存储结构。

感谢聆听